

Hours of Operation (HOO)

The hours of operation definitions in GDT products are based on the following European Geographic Data Files (GDF) Standard version 3.0, Appendix A, section 1.15 "Syntax for Time Domains".

General description

A basic Time Domain is the combination of a Starting Date and a Time Duration with the following notation:

[(Starting Date) { Time duration }].

For example, [(M5d1){d1}] means :

- + Starting Date : any year, month 5 (May), day 1st, at 0:00am.
- + Duration : 1 complete day (i.e. 24 hours or 1440 minutes).

Starting Date syntax

Introduction

Starting Dates are defined by means of a set of graphical symbols allowing the description of years, months, weeks, days, and so on down to the smallest time unit which is the second. The symbols have to be organized in a sequential order starting with the longest time unit. Valid symbol combinations are shown in section 9.1.3.

Each particular symbol is composed of a time type code which indicates a particular time unit (e.g. y for year) and a certain number of digits which represent the time values (up to 4 digits).

Year

ynnnn Defines a particular year. Eg., (y1991) means the year 1991. When no more information is given, (y1991) means 1st of January 1991 at 0:00:00 am.

Month

Mnn Defines a particular month within a particular year, or any year when no "y" information is given. The domain runs from 1 to 12, meaning January and December respectively. (M5) means every 1st of May at 0:00:00 am, whatever the year may be.

Week

wnn Defines a week within a previously defined year, or any year when no "y" information is given. The domain runs from 1 to 53 indicating week number 1 and week number 53 respectively.

Day

Four different time type codes for a "Day" are defined. The code which has to be used depends on whether a particular day in a month, a particular day of the week or a day in a particular week of a month needs to be represented.

dnn Defines a particular day within a particular month if previously defined with the "M" format. When no month information is given, (dnn) means the nnth day in any month. For example, (...d14) means the 14th day in the previously defined month (if any), in the previously defined year (if any) at 0:00:00 am. The domain runs from 1 to 28,29,30 or 31, depending on the month.

tn Defines a particular weekday in a previously (if any) defined week. Domain of values is the following: 1:Sunday, 2:Monday, 3:Tuesday, 4:Wednesday, 5:Thursday, 6:Friday, 7:Saturday. For example, (M5t2) means each Monday in the 5th month (May) of any year, at 0:00:00 am.

fxn Defines a particular weekday in a previously (if any) defined month, with the following rules: n is used as in the "t" format with the same domain of values, 1:Sunday up to 7:Saturday. For "x" one of the following values has to be substituted: 1:first, 2:second, 3:third, 4:fourth, 5:fifth. For example: (...f12) means the first Monday at 0:00:00 am.

lxx Defines a particular weekday in a previously (if any) defined month, with the following rules: n is used as for the "t" format with the same domain of values, 1:Sunday up to 7:Saturday. x has to be chosen from the following set: 1:first, 2:second, 3:third, 4:fourth, 5:fifth. Values are given in reverse order starting from the end of the month and have the meaning: 1=last, 2=last but one, 3= last but two, etc. For example, (...l12) means the last Monday at 0:00:00 am.

Hour

hnn Defines a particular hour within a particular day (if previously defined). When no day is specified, it means that every day is valid. The domain runs from 0 to 23. For example, (d12h6) means every 12th day of a month at 6:00:00 am.

Minute

mnn Defines a particular minute within a particular hour (if previously defined). When no hour is defined, it means that any hour is valid. The domain runs from 0 to 59. For example, (d12h6m30) means every 12th day of a month at 6:30:00 am.

Second

snn Defines a particular second within a particular minute (if previously defined). When no minute is specified, it means that any minute is valid. nn domain is from 0 to 59. For example, (d12h6m30s52) means every 12th day of a month at 6:30:52 am.

Summarizing table of all symbols

time unit	reference time frame	notation	value domain	explanation of values
Year		ynnnn	0...9999	any year
Month	in a Year	Mnn	1...12	January, February, etc. to December
Week	in a Year	wnn	1...53	
Day	in a Month	dnn	1...28/29/30/31	Maximum value depends on the month
Day	of the Week	tn	1...7	Sunday to Saturday
Weekday	of a particular Week	fxn	x: 1...5	first, second, etc.
	of a Month		n: 1...7	week of the month Sunday to Saturday
Weekday	of a particular Week	lxn	x: 1...5	last, last but one, etc. week of the month
	of a Month		n: 1...7	Sunday to Saturday
Hour	of the day	hnn	0...23	24 hours format
Minute	of an hour	mnn	0...59	
Second	of a minute	snn	0...59	

Valid format combinations and default values for Starting Dates

General aspects of the combination of Starting Date formats

Starting Dates which are composed of several time units (e.g. 14th of November 1991) are defined by placing the symbols sequentially in a hierarchical order.

However, some constraints have to be taken into account. The possible combinations possibilities are limited.

Valid symbol combinations are shown in the following figure:

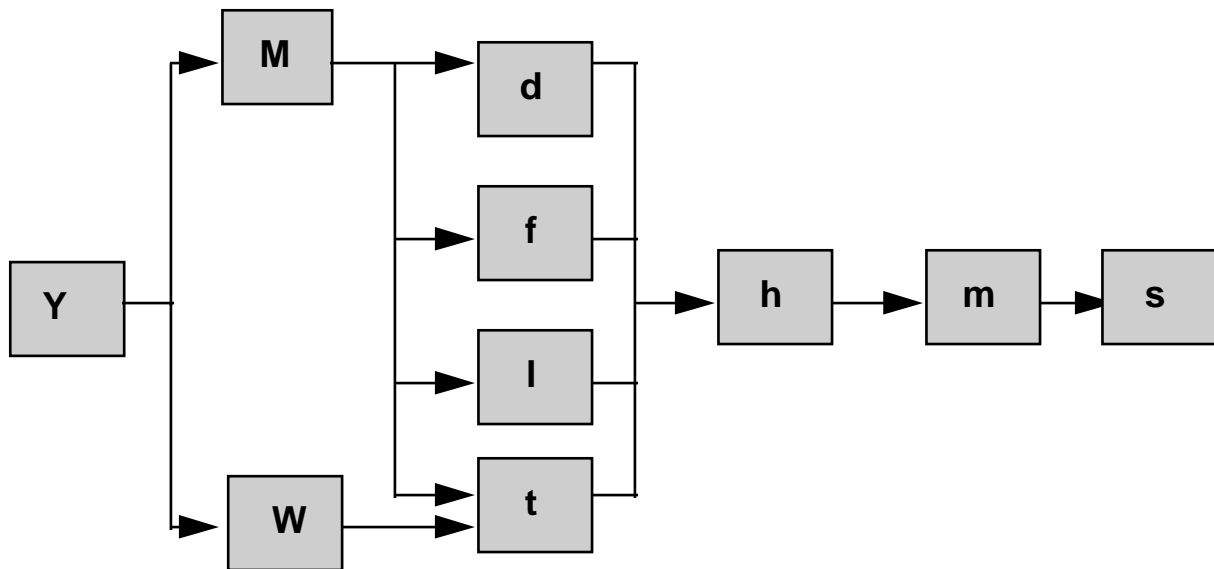


Figure A1.15.1 Valid symbol combinations of starting dates

If not all time type codes of a format combination are used (e.g. only a week and an hour are specified), default values will be adopted for the undefined time units.

General rule for default values

To find the default value for a particular time unit that is undefined in a

Starting Date sequence, a distinction has to be made whether the missing time type code is a "final" one or not.

- "Final" time type Codes: For all time type codes that are missing at the end of a sequence, the default value is the lowest possible value (i.e. M1, w1, d1, h0, m0, s0). If no "Day" is specified (i.e. none of the "d,t,f,l" formats are used) within a defined year and/or month, the default time code and the default value is d1
- "Other" time type codes: If time type codes are missing at the beginning of a sequence or between defined time units, this means that all values of the missing time code are valid.

Examples for default values of undefined time units:

(y1994t1) means:

year 1994, any month/any week, each Sunday (t1), time 00:00:00 am

(w9h11m30) means:

9th week of any year, any day in this week, 11.30 am, 0 sec.

(M4) means:

any year, 1st of April, time 00:00:00 am

(M4m33) means:

any year, April, any day of April, any hour at 33 minutes and 0 seconds.

Detailed description of possible combinations and default values

- y : If no additional "M,w,d,t,f,l,h,m,s" information is given, the default value is M1d1h0m0s0 for the 1st of January at 0:00:00 am in the defined year.
- M : If no additional "y" information is given, it means that any year is valid. If no additional "d,t,f,l,h,m,s" information is given, the default value is d1h0m0s0 for the 1st day in the defined month at 0:00:00 am. No "w" format can be used in combination with the "M" format.
- w : If no "y" extra information is given, it means any year is valid. If no "t,h,m,s" extra information is given, the implicit value is t1h0m0s0 for Sunday in the defined week at 0:00:00 am. No "M,d,l,f" format can be used in combination with the "w" format.
- d : If no additional "y" and/or "M" information is given, it means that any month and/or any year is valid. If no additional "h,m,s" information is given, the default value is h0m0s0 for 0:00:00 am on the defined day. No "w,t,l,f" format can be used in combination with the "d" format.
- t : If no additional "y" and/or "M" or "w" information is given, it means that any month or any week and/or any year is valid. If no "h,m,s" extra information is given, the default value is h0m0s0 for 0:00:00 am on the defined day. No "d,l,f" format can be used in combination with the "t" format.
- f : If no "y" and/or "M" information is given, it means that any month and/or any year is valid. If no additional "h,m,s" information is given, the default value is h0m0s0 for 0:00:00 am on the defined day. No "w,d,t,l" format can be used in combination with the "f" format.
- l : If no "y" and/or "M" information is given, it means that any month and/or any year is valid. If no "h,m,s" extra information is given, the default value is h0m0s0 for 0:00:00 am on the defined day. No "w,d,t,f" format can be used in combination with the "l" format.

- h : If no "y,M,w,d,t,l,f" information is given, it means that any day is valid. If no "m,s" information is given, the default value is m0s0 in the hour in question.
- m : If no "y,M,w,d,t,l,f" information is given, it means that any day is valid. If no "h" information is given, it means that any hour in the previously defined day is valid. If no "s" information is given, the default value is s0 in the minute in question.
- s : If no "y,M,w,d,t,l,f" information is given, it means that any day is valid. if no "h" information is given, it means that any hour in the previously defined day is valid. if no "m" information is given, it means that any minute in the previously defined hour is valid.

Table of allowed and forbidden format combinations

The following table shows valid combinations of Starting Date formats. For each format A of the first column all possible formats B that can follow in a Starting Date sequence are marked by a '*' in the corresponding line.

For example, (M5w1) week 1 in month 5 (May) is not correct, but (y1991w1) week 1 in year 1991 is allowed.

	B	y	M	w	d	t	f	l	h	m	s
A											
y			*	*	*	*	*	*	*	*	*
M					*	*	*	*	*	*	*
w						*			*	*	*
d									*	*	*
t									*	*	*
f									*	*	*
l									*	*	*
h										*	*
m											*
s											

Starting Date examples

"14th November 1991 (at 0:00:00 am)" :
(y1991M11d14).

"Every 2nd of May at 5:31 pm (any year, default second=00)" :
(M5d2h17m31).

"Each last Sunday in February (any year, at 0:00:00 am)" :
(M2l11).

"Monday in week 41 year 1991 (at 0:00:00 am)" :
(y1991w41t2).

"July 1962 (by default the 1st of July at 0:00:00 am)"
(y1962M7).

Time interval syntax

Introduction

The syntax specified in this section enables the description of intervals by means of a set of symbols representing the time interval units year, month, week, day, hour, minute and second. Attached to a starting date, the interval constitutes a basic Time Domain. Without a starting date, it just indicates a duration.

The symbol is composed of an interval type code which indicates a particular time interval unit (eg. y for year) and up to 2 digits which are destined for the time interval values. If the very first time type code is preceded by a minus sign, it means that the duration is counted in the reverse order.

Years

ynn: Defines a duration of nn years. For example, [(y1991M11d14h5m30s19){y1}] means from 14 November 1991, 5:30:19 am to 14 November 1992, 5:30:19 am. If there is no identical calendar date in the year in question, which occurs only for February the 29th, "plus 1 year" leads to February the 28th of the following year. Notice that {y1} = {M12}.

Months

Mnn: Defines a duration of nn months. For example, [(y1991M11d14h5m30s19){M3}] means from 14 November 1991, 5:31:19 am to 14 February 1992, 5:30:19 am.

If there is no identical calendar date in the target month in question, the last day in this month should become the target calendar day. For example, 31st of January plus 1 month leads to 31st of February, which is not correct. According to the rule mentioned above, 31 January plus 1 month leads to 28 or 29 February depending on the year.

Weeks

wnn: Defines a duration of nn weeks, i.e. nn*7 days. For example, [(y1991M11d14h5m30s19){w2}] means from 14 November 1991 at 5:30:19 am to 28 November 1991, 5:30:19 am. Notice that {w1} = {d7}.

Days

dnn: Defines a duration of nn days, i.e. nn*24 hours. For example, [(y1991M11d14h5m30s19){d2}] means from 14 November 1991 at 5:30:19 am to 16 November 1991, 5:30:19 am. Notice that {d1} = {h24}.

Hours

hnn: Defines a duration of nn hours, i.e. nn*60 minutes. For example, [(y1991M11d14h5m30s19){h10}] means from 14 November 1991, 5:30:19am to 14 November 1991 at 3:30:19 pm. Notice that {h1} = {m60}.

Minutes

mnn: Defines a duration of nn minutes, i.e. nn*60 seconds. For example, [(y1991M11d14h5m30s19){m11}] means from 14 November 1991 at 5:30:19am to 14 November 1991 at 5:41:19 am. Notice that {m1} = {s60}.

Seconds

snn: Defines a duration of nn seconds. For example, [(y1991M11d14h5m30s19){s21}] means from 14 November 1991 at 5:30:19am to 14 November 1991 at 5:30:40 am. Notice that {m1} = {s60}.

Summarizing table of all symbols

time unit	notation	value domain	substitutions	remarks
Year	ynn	1...99		If there is no existing identical calendar date in the target year, the last day of the target month will be seen as the target calendar day (can occur for starting date February 29th)
Month	mnn	1...99	{M12} = {y1}	If there is no existing identical calendar date in the target month, the last day of this month will be seen as the target calendar day.
Week	wnn	1...99		
Day	dnn	1...99	{d7} = {w1}	
Hour	hnn	1...99	{h24} = {d1}	
Minute	mnn	1...99	{m60} = {h1}	
Second	snn	1...99	{s60} = {m1}	

Valid format combinations and default values for Time Durations

Combination of periods

Time periods which are composed of several time interval units are represented by listing the individual symbols sequentially in hierarchical order:

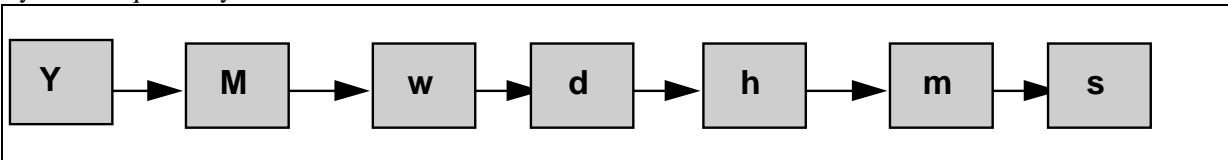


Figure A1.15.2 Combination of periods

The total time duration of a particular combination of symbols is the sum of all single time intervals. For example, {y2M1w2} means a period of 2 years, 1 month and 2 weeks.

Default values

The default value for each missing time value type in a sequence of basic time intervals is 0 (zero).

Time Domain examples

"From 9am to 1pm every day"

Starting date is any year, any month, any day, at 9:00:00 am

(h9)

Duration is 4 hours.

{h4}

So that the complete expression is :

[(h9){h4}]

"From 19:30 to 22:00 every Friday in March"

Starting date is any year, March, any Friday at 7:30 pm

(M3t6h19m30)

Duration is 2 hours and 30 minutes.

{h2m30}

So that the complete expression becomes:

[(M3t6h19m30){h2m30}]

"Last 5 minutes before New Year 1992"

Starting date is the 1st of January 1992 at 0:00:00 am

(y1992) implies the 1st of January at 0:00:00 am

Duration is minus 5 minutes.

{-m5}

So that the complete expression becomes:

[(y1992){-m5}]

Time Domain combinations

General aspects

Since Time Domains can be considered as a set of the smallest time unit described here, the second, Time Domains may also be combined with set operations, such as:

Union of sets	notation : +
Intersection of sets	notation : *
Subtraction of sets	notation : -

Example

A shop is assumed to be "Open to all users": From 9:00am to 12:00am plus from 13:30 to 19:00 each day from Monday to Saturday, except each 1st of May, last Tuesday of January for inventory reasons, and during August (holidays).

The way to code this is to attach the attribute "Opening Period" to the feature "Shopping Centre". The Opening Period refers by means of a Time Domain ID to a corresponding Time Domain Record that contains all opening information.

Because of Morgan's theorem, $A * (B + C) = (A * B) + (A * C)$, there are many different symbol combinations to represent the same complex Time Domain.

The example described above can be solved by the following combination of basic Time Domains :

"From 9:00am to 12:00am" is [(h9){h3}]

"From 13:30 to 19:00" is [(h13m30){h5m30}]

"From 9:00am to 12:00am and From 13:30 to 19:00" becomes:

[[(h9){h3}] + [(h13m30){h5m30}]]

Since this is valid only from Monday to Saturday, an intersection operation is required with the Time Domain "Any week from Monday to Saturday", represented by [(t2){d6}]

The expression now becomes : [[[(h9){h3}] + [(h13m30){h5m30}]] * [(t2){d6}]]

We will now deal with the restrictions :

"1st of May every year", which is represented as [(M5d1){d1}]

"last Tuesday of January" which is represented as [(M11t3){d1}]

"All days during August" which is represented as [(M8){M1}]

The final expression becomes then :

[
 [[[(h9){h3}] + [(h13m30){h5m30}]] * [(t2){d6}]]
 -[(M5d1){d1}]
 -[(M11t3){d1}]
 -[(M8){M1}]
].

Resolution of a Time Equation

Introduction

The problem is to know whether a particular moment (second) belongs to a given Time Domain or not. When the moment in question is within that Time Domain, the boolean value True is attached to the Time Domain.

If not, the boolean value evaluates to False.

Boolean tables

"*" is the boolean AND operator, "+" is the OR operator, and "-" is the "A AND NOT B" operator.

The boolean tables for Time Domain combinations are :

$A + B$	B	<i>True</i>	<i>False</i>
A			
True		T	T
False		T	F

$A * B$	B	<i>True</i>	<i>False</i>
A			
True		T	F
False		F	F

$A - B$	B	<i>True</i>	<i>False</i>
A			
True		F	T
False		F	F

Example of a resolution

Assume we want to know if the previous shop is open on 14 November 1991 at 10:20 am. We have to check if this particular moment fits with the Time Domain where the attribute Opening Period refers to.

14 November 1991, 10:20 am matches the following basic domains:

y1991 / M11 / w46 / d14 / t5 / f25 / l25 / h10 / m20 / s0

A check of the defined Time Domain results in:

"From 9:00 am to 12:00am": [(h9){h3}] is True

"From 13:30 to 19:00" : [(h13m30){h5m30}] is False "From Monday to Saturday": [(t2){d6}] is True

Therefore the expression [[[(h9){h3}] + [(h13m30){h5m30}]] * [(t2){d6}]] is True

"1st of May every year": [(M5d1){d1}] is False

"last Tuesday of January": [(M11t3){d1}] is False "All during August": [(M8){M1}] is False

Thus the complete expression

```
[
  [ [ [(h9){h3}] + [(h13m30){h5m30}] ] * [(t2){d6}] ] -[(M5d1){d1}]
  -[(M11t3){d1}]
  -[(M8){M1}]
]
```

evaluates to True: The shop is open.